

USB for a 1977 Keyboard



The Vista80 Keyboard



About Angel

- Hardware Hacker
 - Programmer
 - Lives in Montreal, Canada
 - Loves Hacking Old Stuff



Vista80 Character Generator



- 8080A processor
- Dual 8 Inch Floppies



House of Commons Video



Vista80 Brochure



Features for
Broadcast Application



Election Results
Displayed



Advanced All Solid State
Software Controlled
Systems



Special 3-D and
Color Effects



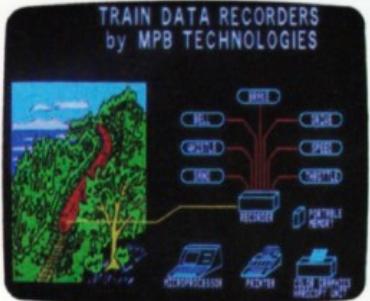
Large Selection of
Font Styles and Sizes



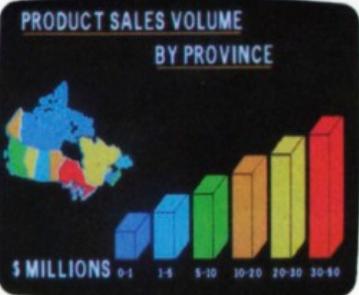
Weather Maps



Vista80 Brochure



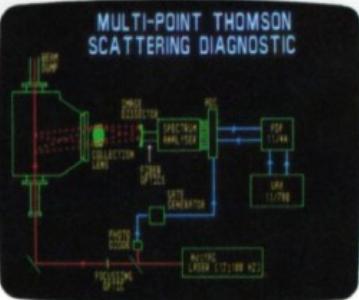
Technical Information



Charts and Illustrations

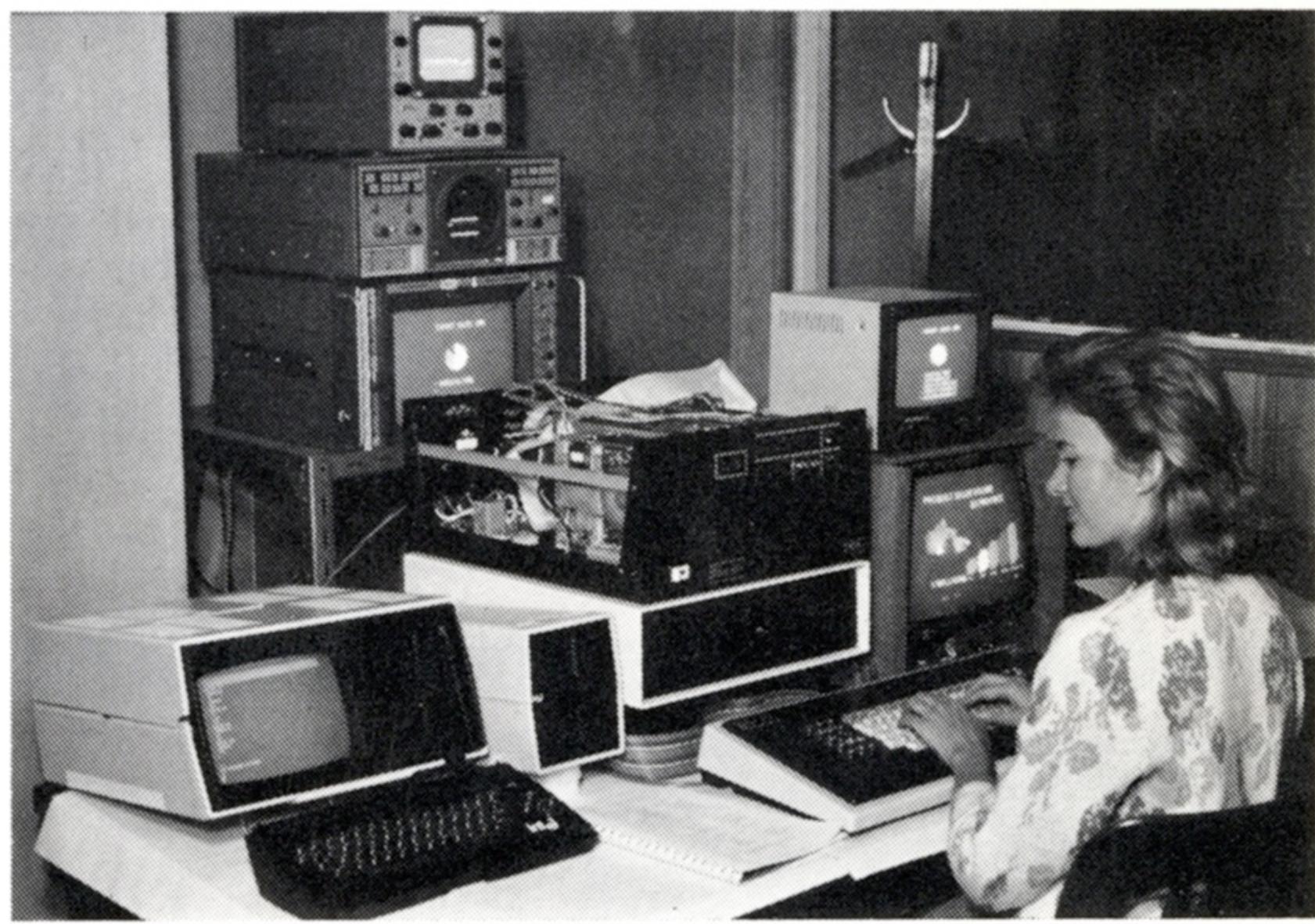


Logos and Special Effects

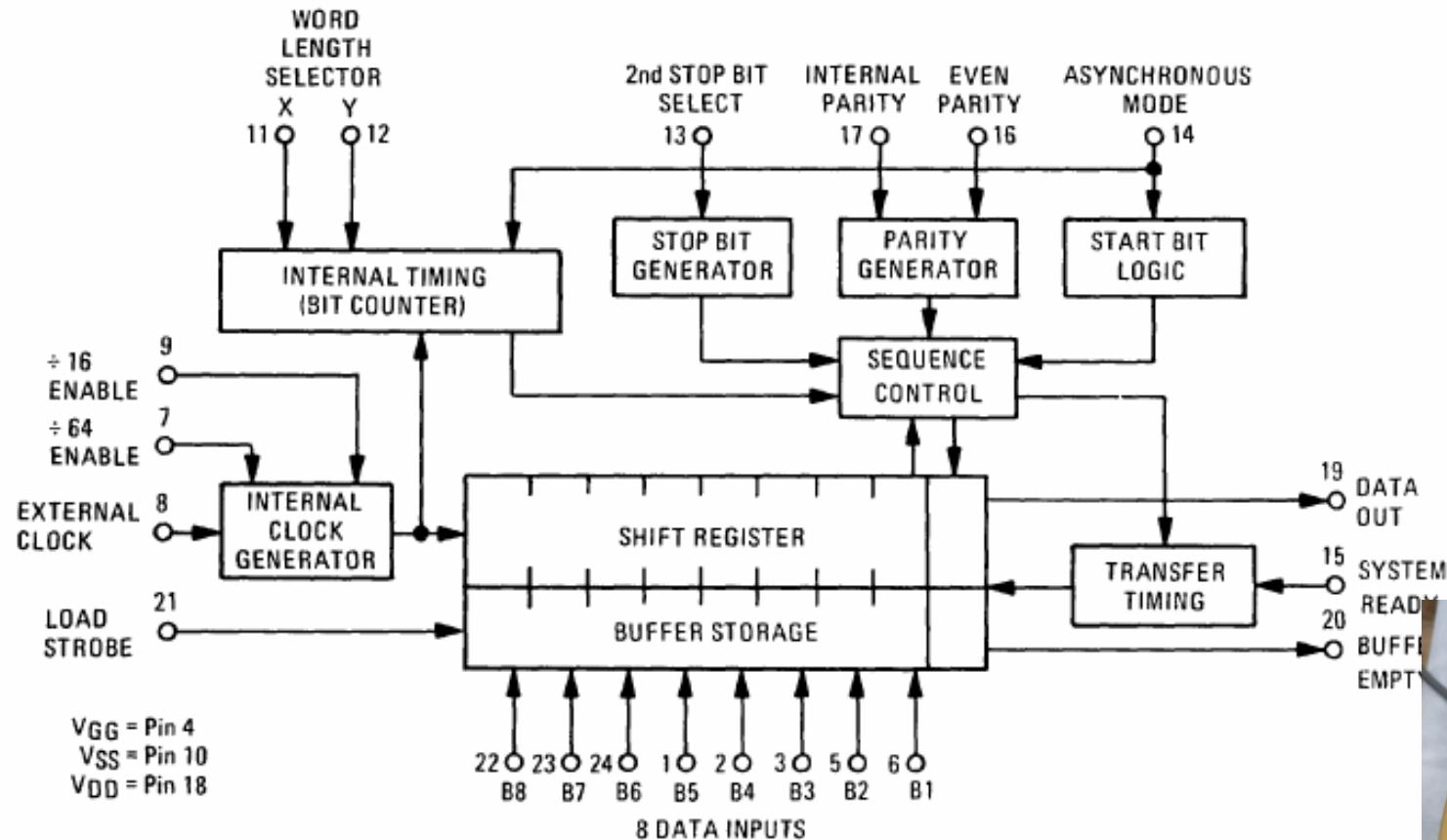


Schematic Diagrams





MC2257L
Terminal Transmitter



Future USB Keyboard



13



Teensy 2.0

- Made by PJRC in the USA
- Based on the ATMEGA32U4
- 16MHz
- 32 KB of Flash
- 2.5 KB of SRAM
- 1KB of EEPROM

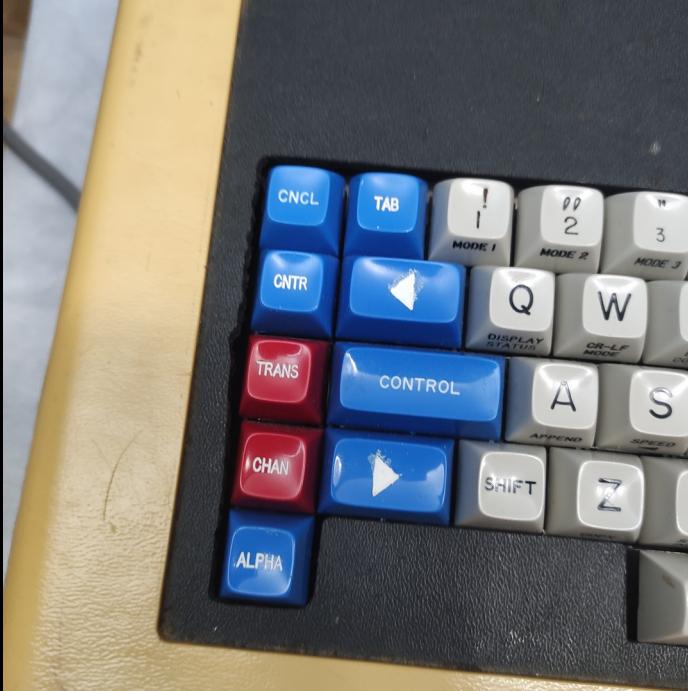


Test Code

```
bool readed = false;
void loop() {
    // put your main code here, to run repeatedly:
    bool pinState = digitalRead(0);
    if(!pinState && !readed){
        byte portState = PINB;
        Serial.println(portState, BIN);
    }
    if(pinState && readed){
        readed = false;
    }
}
```



Special keys



Character Tables

```
//LCL          ` a   b   c   d   e   f   g   h   i   j   k   l   m   n   o
const char Lowr1[16] = {0x60,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69,0x6A,0x6B,0x6C,0x6D,0x6E,0x6F};
//LCU          p   q   r   s   t   u   v   w   x   y   z   {   _   }   ~   CNTR (Ctrl)
const char Lowr2[16] = {0x70,0x71,0x72,0x73,0x74,0x75,0x76,0x77,0x78,0x79,0x7A,0x7B,0x7C,0x7D,0x7E,0x00};

//HCL          @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
const char Uppr1[16] = {0x40,0x41,0x42,0x43,0x44,0x45,0x46,0x47,0x48,0x49,0x4A,0x4B,0x4C,0x4D,0x4E,0x4F};
//HCU          P   Q   R   S   T   U   V   W   X   Y   Z   [   \   ]   ENDL ENDP
const char Uppr2[16] = {0x50,0x51,0x52,0x53,0x54,0x55,0x56,0x57,0x58,0x59,0x5A,0x5B,0x5C,0x5D,0x5E,0x00};

//NUM          0   1   2   3   4   5   6   7   8   9   :   ;   <   =   >   ?
const char Numbr[16] = {0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3A,0x3B,0x3C,0x3D,0x3E,0x3F};
//SYM          SPC   !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
const char Symb1[16] = {0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x2A,0x2B,0x2C,0x2D,0x2E,0x2F};

// Binding:           SUPR TAB  ESC  DEL           Insrt      PGUP  FMOD  PGDN  BKSP
//                   BLUE TAB CNCL ERAP ERAL PINK C+P  LBLU WHIT INSR COLR BLFT TRAN BRGT RTN  CTRL+CNCL
const char Ctrl1[16] = {0x00,0xB3,0xB1,0xD4,0x00,0x00,0x00,0x00,0xD1,0x00,0xD3,0x00,0xD6,0xB2,0x00};

// Binding:           SHFT      HOME     ALT           RGHY      DOWN      ENTR LEFT   UP
//                   CHAN C+Q  HOME C+C  ALPA BLAK RED  GREN RGHt C+Y  DOWN C+K  YELO  CR  LEFT   UP
const char Ctrl2[16] = {0x00,0x00,0xD2,0x00,0x00,0x00,0x00,0xD7,0x00,0xD9,0x00,0x00,0xB0,0xD8,0xDA};
```



Project requirements

- ATTINY
- Only minimum needed
- 16 IO
- Supports VUSB

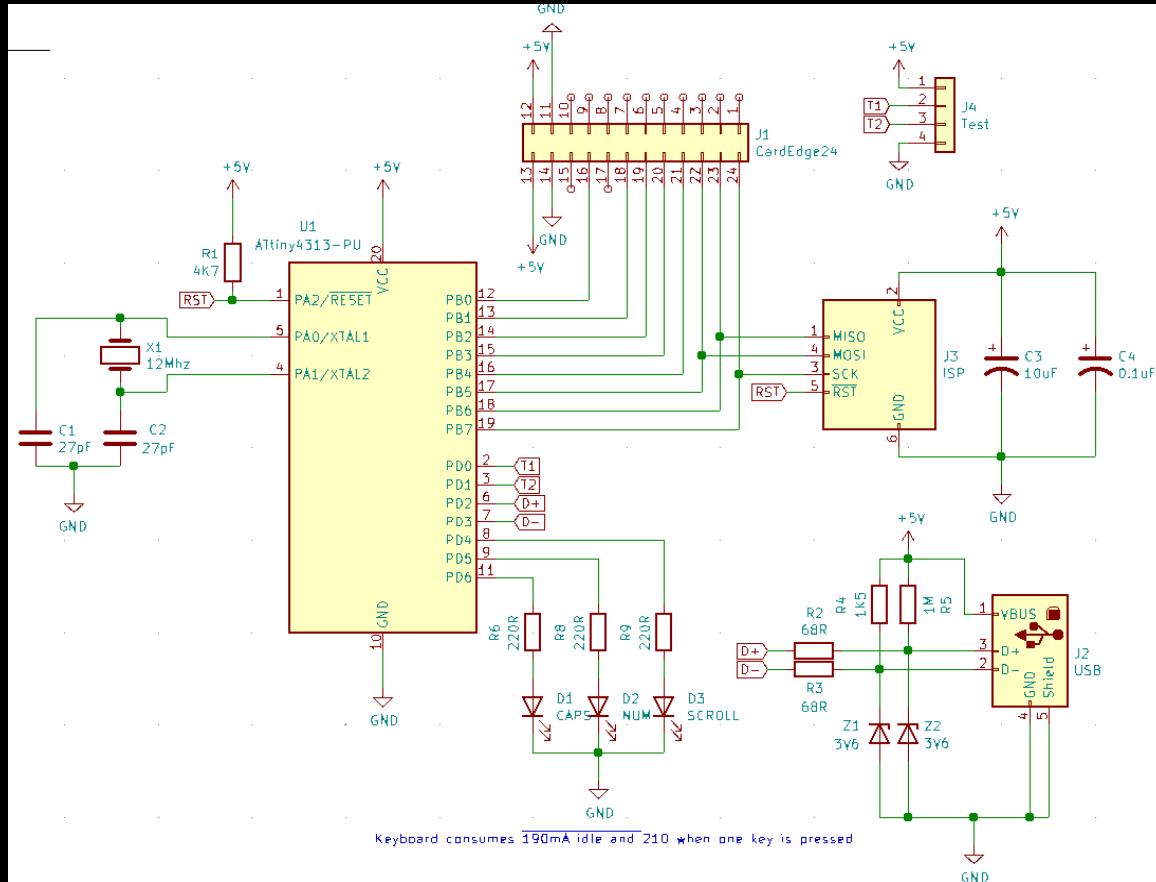


ATTINY4313

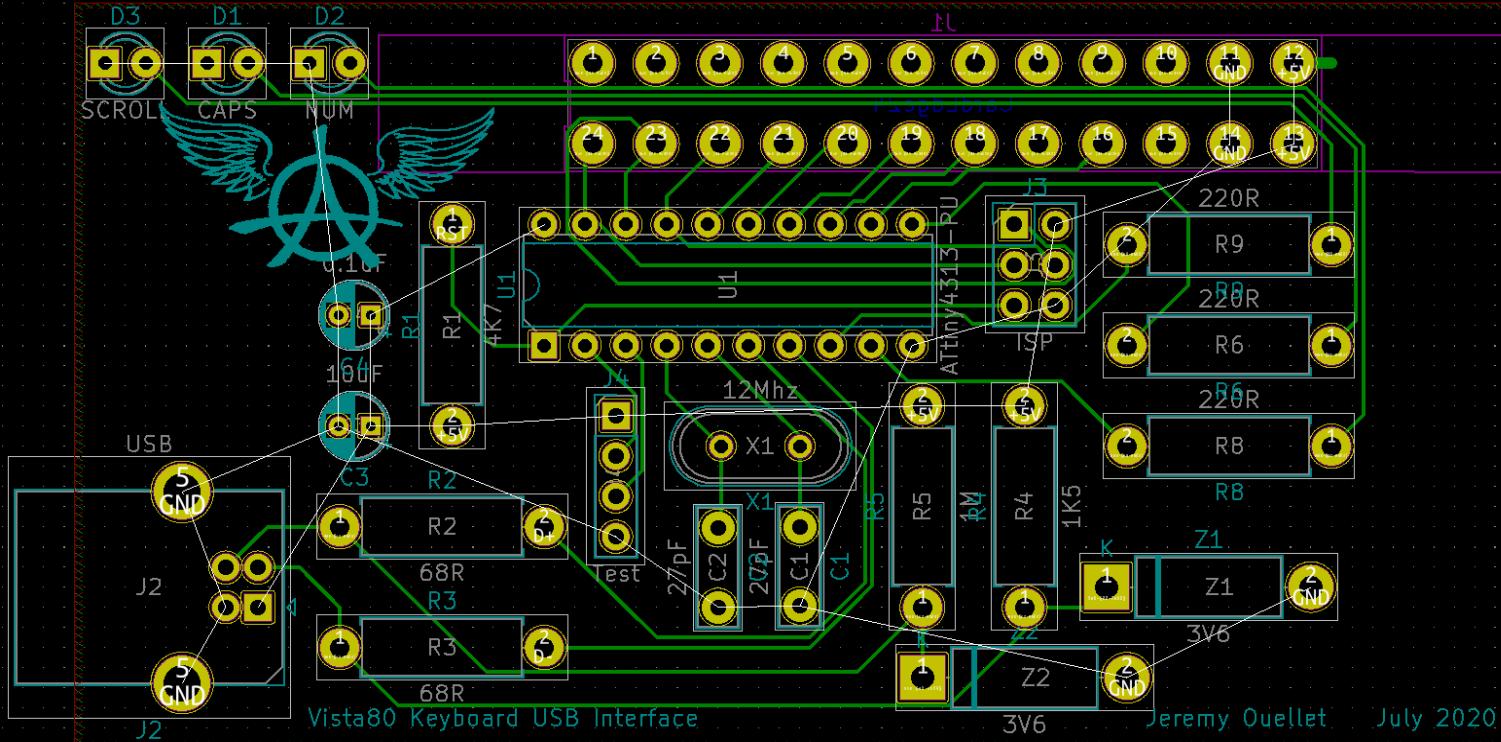
- RISC Processor
- 4K Flash
- 256 Bytes Ram
- 256 Bytes EEPROM
- 18 IO



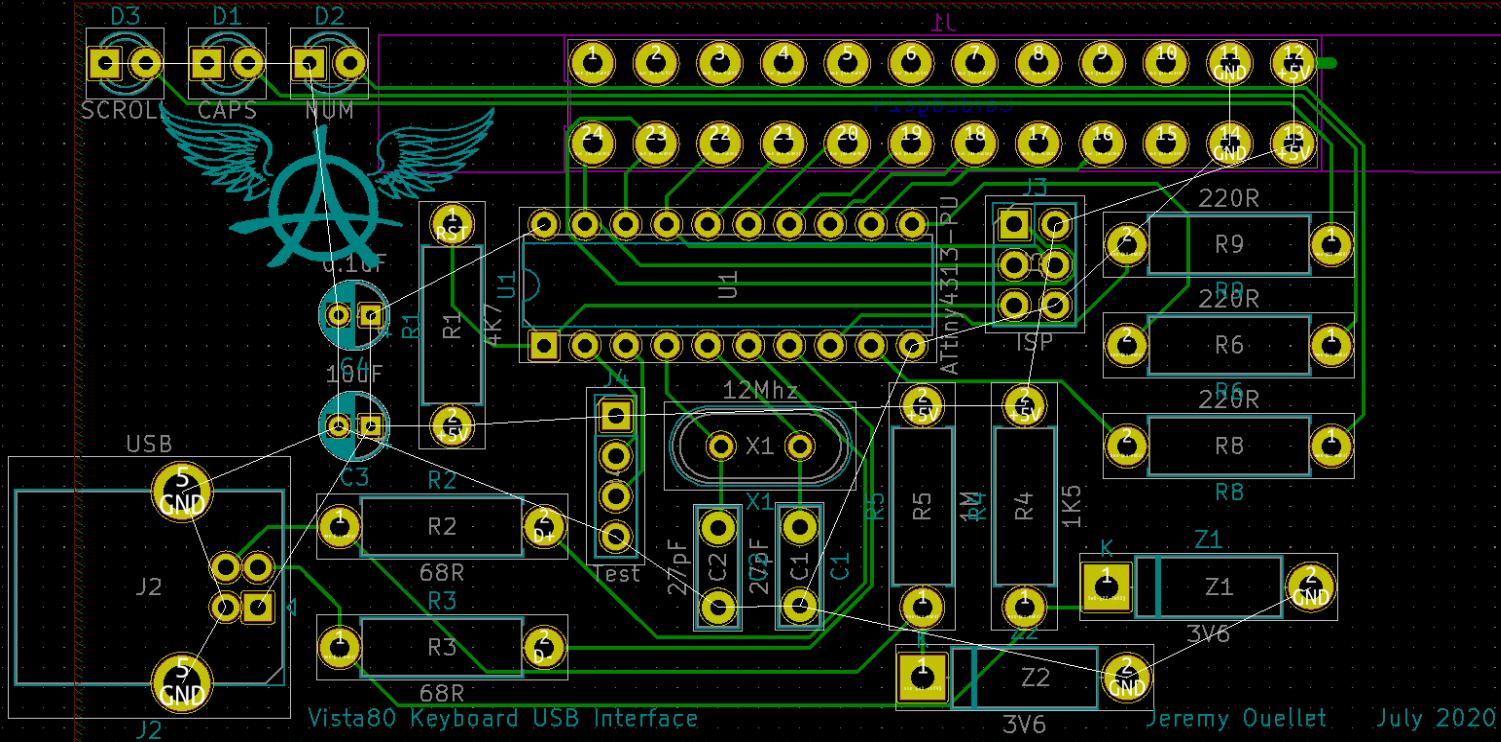
Schematic



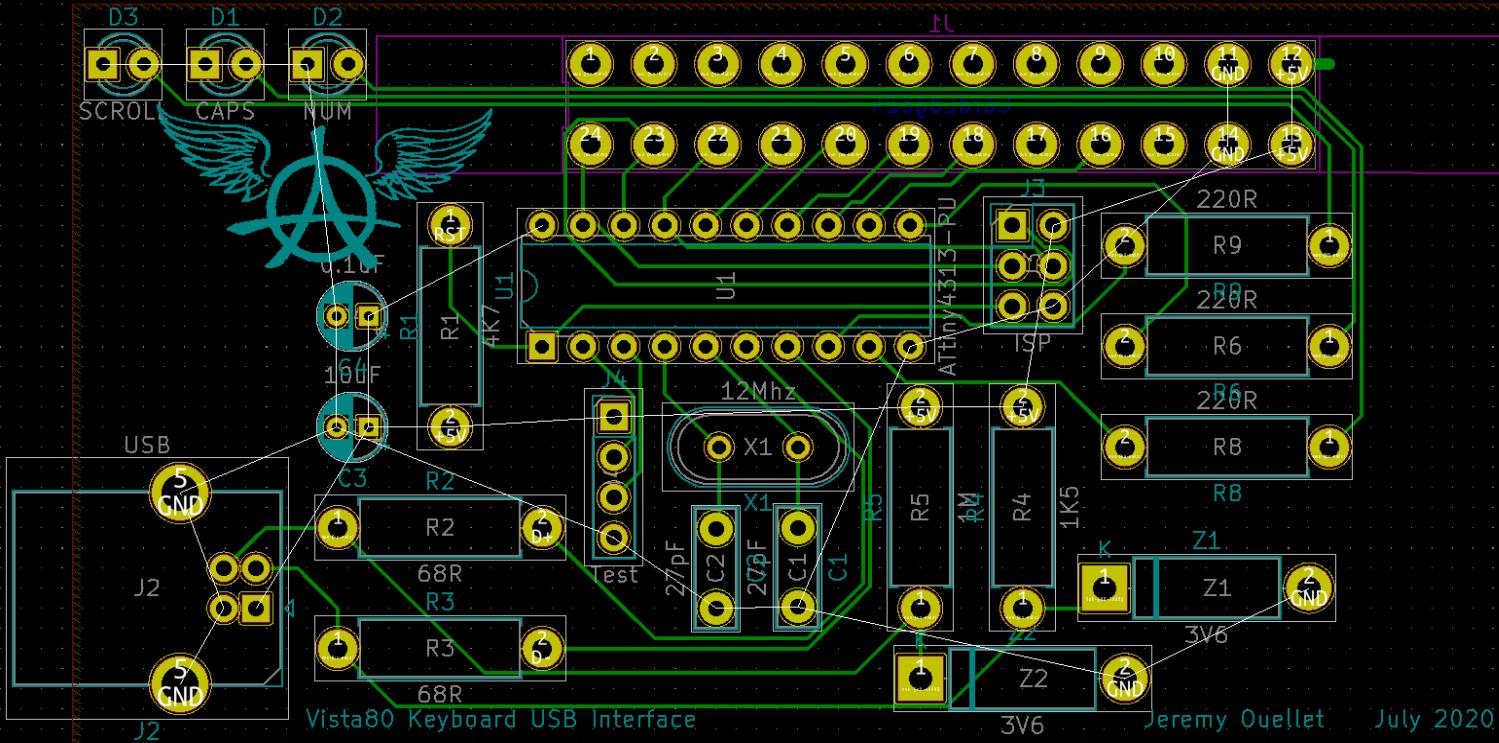
PCB



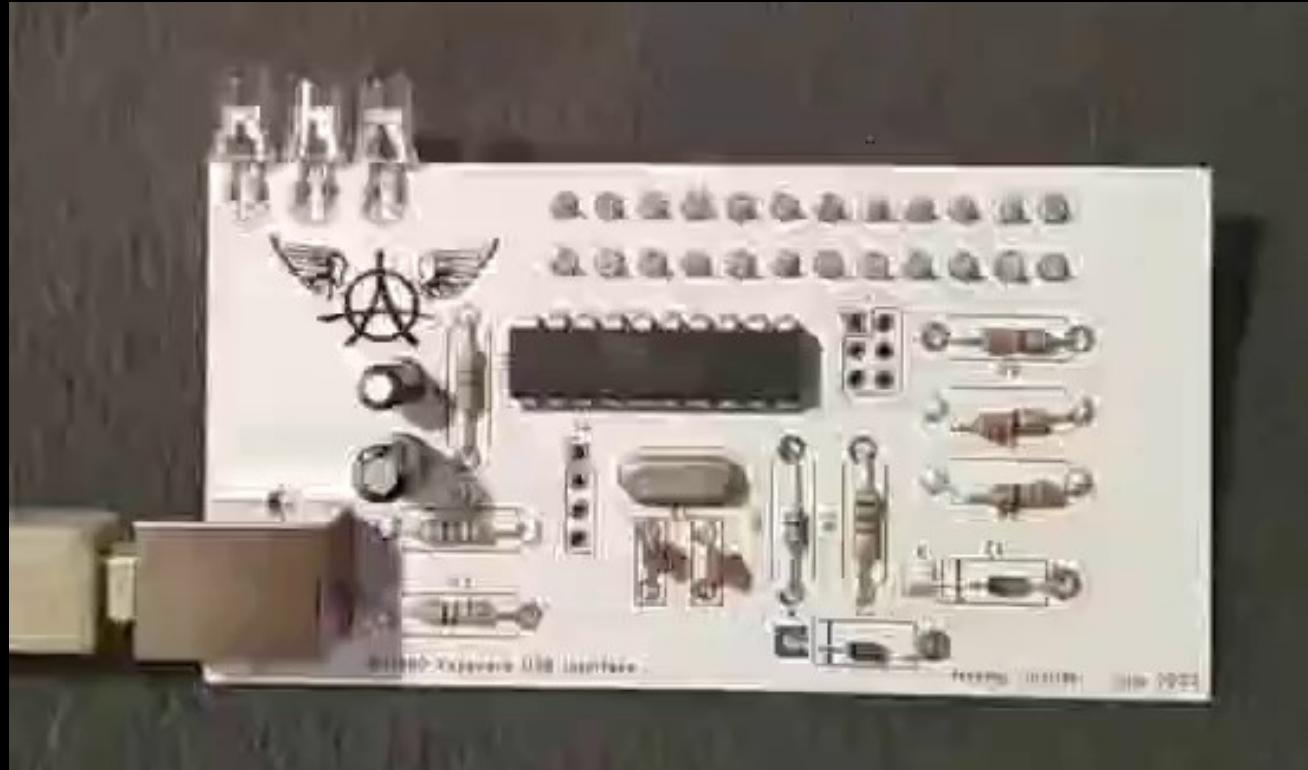
PCB



PCB



LED_BLINK



:D



C64 USB



Serial Output

```
if(!pinState && !readed){  
    byte portState = PINB;  
    bool ctrl = !((portState >> 6) & 0x01);  
    bool shft = !((portState >> 5) & 0x01);  
    bool othr = !((portState >> 4) & 0x01);  
    byte all = ~(portState >> 4) & B0111;  
    byte key = portState&B00001111;  
    if(ctrl){Serial.print("CTRL ")}  
    else{Serial.print("   ")}  
    if(shft){Serial.print("SHFT ")}  
    else{Serial.print("   ")}  
    if(othr){Serial.print("OTHR ")}  
    else{Serial.print("   ")}  
    Serial.print("PORTB: ");  
    Serial.print(portState);  
    //Serial.print(" CHAR: ");  
    //Serial.print((portState&B00011111 + 64));  
    Serial.print(" ALL: ");  
    Serial.print(all, BIN);  
    Serial.print(" BIN: ");  
    Serial.println(portState, BIN);
```



Connected Code

```
//send packet every so often to make sure we are still connected
if (sendCounter++ == 10000) {
    > byte tmp[1] = {0};
    > usbSetInterrupt(tmp, 1);
    > sendCounter = 0;
}

if (connected) {
    > setLeds(ledState);
    > updateNeeded = readPort();
    > /* If an update is needed, send the report */
    > if (sendReport(updateNeeded)) {
    >     > updateNeeded = false;
    >     > resetNeeded = true;
    > }
    > //If the package has been sent and the keys need clearing, clear them.
    > if (resetKeys(resetNeeded)) {
    >     > resetNeeded = false;
    > }
} else {
    > //not connected, flash the leds
    > if (flashCounter++ == 10000) {
    >     > PORTD ^= (1 << PORTD4);
    >     > PORTD ^= (1 << PORTD5);
    >     > PORTD ^= (1 << PORTD6);
    >     > flashCounter = 0;
    > }
}
```



Size of code

```
make all
Invoking: Print Size
avr-size --format=avr --mcu=attiny4313 Vista-Test.elf
AVR Memory Usage
-----
Device: attiny4313

Program:      3746 bytes (91.5% Full)
(.text + .data + .bootloader)

Data:         132 bytes (51.6% Full)
(.data + .bss + .noinit)
```



Thanks

- Canadian Science and Technology Museum
- MPB Communications
- Mikkel Holm Olsen
- KiCad
- Friends & Partners



Contact Info

- Email: angel@jouellet.net
- XMPP: angel@jouellet.net
- Website: <https://jouellet.net>
- Call Sign: KC3NZD

